

B. REMARKS

The Examiner is thanked for the performance of a thorough additional search. In this Reply, Claims 18-24 have been amended. Hence, Claims 1-24 are pending in this application. The amendments to the claims do not add any new matter to this application. Furthermore, the amendments to the claims were made to correct minor typographical errors and not for any reason related to patentability. All issues raised in the Office Action mailed December 5, 2003 are addressed hereinafter.

OBJECTION TO CLAIMS 18-24

Claims 18-24 were objected to on the ground that the term “computer-readable medium” has not been recited in and is not consistent with the terminology of base Claim 17. Claims 18-24 have been amended herein to change the term “computer-readable medium” to “apparatus.” Accordingly, reconsideration and withdrawal of the objection to Claims 18-24 is respectfully requested.

REJECTION OF CLAIMS 1-24 UNDER 35 U.S.C. § 103(a)

Claims 1-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Gorshkov et al.*, U.S. Patent No. 6,490,721 (hereinafter “*Gorshkov*”) in view of *Bowman-Amuah*, U.S. Patent No. 6,442,748 (hereinafter “*Bowman*”) and further in view of *Khoyi et al.*, U.S. Patent No. 5,303,379 (hereinafter “*Khoyi*”). It is respectfully submitted that Claims 1-24 are patentable over *Gorshkov*, *Bowman* and *Khoyi* for at least the reasons provided hereinafter.

CLAIM 1

Claim 1 recites a method of debugging a first software program that requires:

“preserving a memory state of a preserved portion of the first software program;
dynamically linking a second software program to the first software program
without deallocating from volatile memory the first software program;
executing the second software program; and
if execution of the second software program would otherwise cause modification
to targeted data that is in the preserved portion of the first software
program, then making a copy of the targeted data and modifying the copy
of the targeted data to generate a modified copy of the targeted data
without modifying the targeted data that is in the preserved portion of the
first software program.”

It is understood from the Office Action that the parent process described in *Gorshkov* is considered to be the “first software program” recited in Claim 1 and that the child process is considered to be the “second software program” recited in Claim 1. *Gorshkov* describes that the parent process consists of the target program to be debugged and the debugging user actions needed from the user action libraries (Col. 3, lines 45-53). *Gorshkov* also describes that the child process is created using a fork process and is a copy of the parent process.

In operation, the child process attaches to and patches into the parent process by inserting calls to debugging routines into the parent process. More specifically, the child process inserts, into the memory image of the target program contained in the parent process, calls to user actions contained in the user action library. The child process allocates space for the patch in a patch area in the parent process and replaces one or more instructions in the target program with a branch instruction to the patch area. The child process also generates code in the patch area to call the user action, i.e., the debugging routine (Col. 4, lines 1-38). When the modified image of the target program

contained in the parent process is executed, it calls the debugging routines contained in the user action library.

The Office Action admits that *Gorshkov* does not teach or suggest “making a copy of the targeted data and modifying the copy of the targeted data to generate a modified copy of the targeted data without modifying the targeted data that is in the preserved portion of the first software program,” since given that the portion of the parent process duplicated in the child process is considered to be the “copy of the targeted data” recited in Claim 1, *Gorshkov* does not teach or suggest modifying the child process. Furthermore, *Gorshkov* teaches modifying the parent process when the child process patches the debugging routines into the parent process. The Office Action instead relies upon *Bowman* (Col. 177, lines 38-58; Col. 293, lines 32-56; and FIGS. 177, 178) and *Khoyi* (Col. 3, lines 33-44 and Col. 38, lines 7-22) for these limitations.

The text at Col. 177, lines 38-58 of *Bowman* describes a software configuration management solution that allows developers to make changes to an open edition or checked out copy of a class to maintain version control while increasing development bandwidth. The text at Col. 293, lines 32-56 of *Bowman* describes maintaining consistency of data by assigning each logical unit of work independent copies of a portion of a business model to prevent transactions from interfering with each other.

The text at Col. 3, lines 33-44 of *Khoyi* describes an approach for creating new objects by making copies of a prototype copy of an object. The copies of the prototype copy of an object may then be modified by users without affecting the prototype copy of the object. The text at Col. 38, lines 7-22 of *Khoyi* describes making copies of linked

objects where in some situations a link is copied and in other situations linked data is copied. Also, copies of objects may be modified independently of the original version.

It is respectfully submitted that the invention recited in Claim 1 is not obvious in view of *Gorshkov*, *Bowman* and *Khoyi* because modifying the software debugging method of *Gorshkov* with the approaches described in *Bowman* and *Khoyi* would render the software debugging method of *Gorshkov* unsatisfactory for its intended purpose. Both *Bowman* and *Khoyi* involve making copies of objects and then updating only the copies and not the original data. Applying this approach to the software debugging method of *Gorshkov* would involve not updating the target program contained in the parent process. This would mean that the target program contained in the parent process would not contain calls to the debugging routines contained in the user action library. Without these calls, the debugging routines would not be invoked, which would render the software debugging method of *Gorshkov* unsatisfactory for its intended purpose. Under MPEP § 2143.02 (Page 2100-2127), citing *In re Gordon*, 221 USPQ 1125 (Fed. Cir. 1984), “[i]f proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification.”

It is also respectfully submitted that the combined teachings of *Gorshkov*, *Bowman* and *Khoyi* are not sufficient to render Claim 1 *prima facie* obvious because the proposed combination would change the principle of operation of the software debugging method of *Gorshkov*. As described above, applying the approaches of *Bowman* and *Khoyi* to the software debugging method of *Gorshkov* would involve not updating the target program contained in the parent process, which would not only change the

principle of operation of the software debugging method, but would render the software debugging method completely inoperable. Under MPEP § 2143.02 (Page 2100-2127), citing *In re Ratti*, 123 USPQ 349 (CCPA 1959), “[i]f the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.”

In view of the foregoing, it is therefore respectfully submitted that Claim 1 is not obvious in view of *Gorshkov*, *Bowman* and *Khoyi* and is therefore patentable over *Gorshkov*, *Bowman* and *Khoyi*.

CLAIMS 2-8

Claims 2-8 all depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 2-8 are patentable over *Gorshkov*, *Bowman* and *Khoyi* for at least the reasons set forth herein with respect to Claim 1. Furthermore, it is respectfully submitted that Claims 2-8 recite additional limitations that independently render them patentable over *Gorshkov*, *Bowman* and *Khoyi*.

CLAIMS 9-16

Claims 9-16 recite limitations similar to Claims 1-8, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 9-16 are patentable over *Gorshkov*, *Bowman* and *Khoyi* for at least the reasons set forth herein with respect to Claims 1-8.

CLAIMS 17-24

Claims 17-24 recite limitations similar to Claims 1-8, except in the context of apparatti. It is therefore respectfully submitted that Claims 17-24 are patentable over *Gorshkov*, *Bowman* and *Khoyi* for at least the reasons set forth herein with respect to Claims 1-8.

It is respectfully submitted that all of the pending claims are in condition for allowance and the issuance of a notice of allowance is respectfully requested. If there are any additional charges, please charge them to Deposit Account No. 50-1302.

The Examiner is invited to contact the undersigned by telephone if the Examiner believes that such contact would be helpful in furthering the prosecution of this application.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Edward A. Becker

Reg. No. 37,777

Date: February 4, 2004

1600 Willow Street
San Jose, CA 95125
(408) 414-1204
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Non-Fee Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

On February 4, 2004

By

Angelica Jaquez